

CASE STUDY

Open Access



Creating an integrated collaborative environment for materials research

Matthew D. Jacobsen¹, James R. Fourman², Kevin M. Porter³, Elizabeth A. Wirrig², Mark D. Benedict¹, Bryon J. Foster¹ and Charles H. Ward^{1*}

* Correspondence: charles.ward.4@us.af.mil

¹Materials and Manufacturing Directorate, Air Force Research Laboratory, Wright-Patterson AFB, OH 45433, USA

Full list of author information is available at the end of the article

Abstract

This paper describes the creation of a cyberinfrastructure to facilitate collaborative materials research in a laboratory environment that supports the discovery, development, and sustainment of materials and processing solutions. The infrastructure provides a web-based interface supporting group and project spaces within which researchers can easily organize, share, and collaborate on the results of their experimental and computational efforts. It seamlessly connects researchers with experimental and computational resources for easy generation, collection, and storage of digital data to provide instant access to results with no intermediate transfers. Persistent identifiers and metadata tagging are used to ensure historical research data are discoverable, interpretable, and reusable. The architecture is designed to be modular and agile and is based on federation of both applications and data through a central service bus that brokers all transactions. It is comprised of a number of open-source, commercial, and non-commercial software packages that provide the specific functionality needed to meet the large number of system requirements. This collaborative environment is essential to enabling a large research organization to conduct a research program consistent with the discipline of Integrated Computational Materials Engineering by allowing the seamless connection of experiment to model through pedigreed digital data with complete provenance.

Keywords: Data management, Workflow, Collaboration, Materials Genome Initiative, Integrated Computational Materials Engineering

Background

The discipline of Integrated Computational Materials Engineering (ICME) defines the need to capture and integrate materials information for reuse to enhance scientific and engineering efficiency and provide new pathways for discovery and development [1]. This concept is integral to the Materials Genome Initiative's (MGI) call for the creation of a materials innovation infrastructure based on advances in and coupling of experimental tools, computational tools, and digital data [2]. This approach is but a natural evolution of materials science and engineering given the advances in the physical understanding of materials phenomena, experimental techniques, computational power, and maturity in software development and data science. However, if an organization is to capture, integrate, and reuse its materials information, it must have a means to do so efficiently in order to minimize the administrative burden on the researcher while maintaining provenance of the research process. Hence, a software platform that can

support seamless and collaborative materials research enabled by modern web-based interfaces, content management systems, and federated data architectures is essential to facilitate advances in materials discovery, development, and transition in a contemporary paradigm.

The field of chemistry recognized a need for electronic tools to facilitate internal research and technology development in the 1970s when efforts to provide “laboratory automation” resulted in the advent of the Laboratory Information Management System (LIMS) in the early 1980s [3]. Early LIMS development efforts focused on providing a means to store and provide access to well-structured research data. Electronic laboratory notebooks (ELNs) were introduced in the 1990s, also in the field of chemistry, as a means to capture and maintain the unstructured research data typically found in a paper laboratory notebook [4]. Generally, both LIMS and ELN solutions have been developed to meet very specific discipline requirements, primarily in chemistry and biology. The same highly tailored features that make them so useful for a specific discipline also make them difficult to transition between disciplines. As both LIMS and ELN share the mutual goal of maintaining and reusing corporate knowledge to enhance research efficiency, a need for more access to integrated information began to blur the lines between LIMS and ELN in the 2000s.

Circa 2000, a number of efforts began to build software platforms to facilitate broad access to scientific analysis tools and research data using the Internet in the fields of astronomy, plant biology, and nanotechnology, for example [5–7]. Among other features, these platforms have introduced the capability for remote access to simulation codes and high-performance computing resources by researchers. Notable among these for the purposes of this paper is nanoHUB, which has developed a web-based interface to host simulation tools, education, publishing, and collaboration resources for those interested or involved in nanotechnology research [6]. Stemming from the success of nanoHUB, the supporting HUBzero® software platform was made available as an open source code for research collaboration [8]. There are now at least 60 installations of HUBzero® supporting collaborative research across a broad spectrum of technology including earthquake engineering [9], pharmaceutical engineering and science [10], biomedical research [11], and geospatial research [12].

In materials science and engineering, there has been a sustained emphasis on developing databases and repositories for materials data, particularly for internal engineering application but more recently for scientific uses as well [13–15]. Some of these data management solutions have features that approach a LIMS, but none are sufficient to provide a complete cyberinfrastructure that connects people with equipment, simulation code, high-performance computing, and historical research data in a laboratory environment. The problem is only amplified when a solution is sought in a laboratory setting that is concerned with a multitude of materials classes and disparate applications.

Case description

The Materials and Manufacturing Directorate of the Air Force Research Laboratory has a broad ranging mission to provide materials and manufacturing solutions across the entire life cycle of aerospace materials. Over 700 scientists and engineers are engaged in efforts ranging from basic research to materials and process development, manufacturing scale-up, and support for fielded systems. The scope of research covers

a very wide range of materials classes including semiconductors, ceramics, metals, polymers, composites, and biomaterials. The organization has embraced the ICME and MGI vision of the future of materials science and engineering, establishing a strategic initiative called Integrated Computational Materials Science and Engineering (ICMSE). A critical component of this initiative is to build a cyberinfrastructure that can better support internal collaborative materials research and engineering while preserving the artifacts of the research process for future reuse. This cyberinfrastructure must be able to connect hundreds of pieces of experimental equipment spread across 175,000+ square feet of laboratory space, a high-performance computing resource, hundreds of desktop computers, and the 700+ scientists and engineers in the laboratory.

The background discussion above highlights many functional attributes one would desire in a cyberinfrastructure that is to serve a materials research laboratory with diverse needs. From a materials researcher's perspective, these functional attributes can be summarized as follows:

- *Collaborative group and project spaces to share data, research notes, tasks, and visualizations*—logically configurable and controllable virtual containers that serve as live ELNs
- *Seamless collection of experimental and simulation data from equipment, operators, and simulation*—direct connection between the software platform and experimental equipment, computational clusters, and high-performance computing resources to facilitate data capture, project alignment, and storage
- *Assured data pedigree, provenance, and discovery*—facilitated metadata assignment to data over a broad set of materials data types that can ensure a researcher, including one who did not generate the data, can find and reuse the data at a later date with high confidence
- *Integrated access to modeling code, data analysis tools, and high-performance computing resources*—an ability to efficiently input project data into a complex simulation or analysis, automatically manage the simulation workflow, and couple the results to a workspace that maintains continuity in the research project
- *Single sign-on, identity management, and role-based access controls to data and components*—allows the researcher access to needed software components while providing updated provenance of the research data regardless of point of entry. Provides for the security and veracity of the information contained within a system containing a mix of data at various levels of analysis and control

As noted by Taylor, no single software solution is likely to meet all the needs of a research organization [4]. An important attribute of any such system must be adaptability to both changing materials requirements and advances in software/computation technology. This is certainly the case for a materials laboratory seeking a solution that can accommodate a wide variety of material systems through all stages of the materials life cycle. With the advent of MGI, there are now several activities currently underway to provide cyberinfrastructure solutions to meet the targeted requirements of materials researchers [16–19]. This paper describes efforts to construct an integrated collaborative environment (ICE) using both readily available software packages combined with tailored software solutions to build a cyberinfrastructure to serve the internal needs of a large materials laboratory.

Discussion and Evaluation

Development of a cyberinfrastructure to meet the functional requirements cited above began with an analysis of alternatives (AoA). The AoA considered a number of factors including license model, ability to meet functional requirements, cost, documentation, support burden, and implementation speed. To minimize development risk while increasing the quality of the product and ensuring greater user adoption, an iterative evaluation, development, and implementation life cycle was employed. Candidate materials research processes were selected that contained one or more of the required functional attributes, and these became system “pilots” to guide analysis and then development. A critical design philosophy was to design the pilot implementation for rapid turnaround of functionality in order to quickly identify risk and resolve implementation obstacles. The first pilot for the ICE project was comprised of four representative activities—one synthesis of material, two material characterization techniques, and one material simulation, as well as a conditional feedback loop. Each of these activities were modeled after actual research processes but had never been assembled coherently as a workflow. A workflow diagram and a full set of use cases were developed in order to communicate the requirements to software vendors providing self-contained, candidate commercial-off-the-shelf (COTS) solutions.

A number of COTS solutions, including both LIMS and ELN, were evaluated in rapid fashion as potential “complete” solutions—the objective being that through heavy configuration, these tools would address the bulk of the required functionalities. However, reviews from stakeholders and growing concerns over scalability and customizability required the development team to begin evaluating a solution including custom in-house development. Consistent with Taylor’s observation, none of the products evaluated provided a suitable stand-alone solution to meet the overall functional attribute requirements. Thus, the architecture focus shifted from a purely COTS acquisition approach to a modular and hybrid model of tightly coupled toolsets that are a combination of commercial, open-source, and in-house developed software. The system is composed of various semi-autonomous sub-components around a common federated core. As such, different toolsets can be integrated to provide specific core capabilities system-wide without overall system redesign.

Architectural solution

Common service bus

The AoA result showed that the preferred solution would need to have a modular and extensible architecture with an ability to integrate a variety of technologies to meet requirements. In order to achieve this, an architecture based on a stable common service bus (CSB) was designed to broker all transactions for the numerous sub-components in the ecosystem, as shown in Fig. 1. This design decision addressed several objectives: reduce the number of custom interconnections between sub-components, ensure consistency of data transmission, foster identity management for all entities and objects in the system, and, finally, to enable otherwise self-governed systems to participate in the ICE ecosystem. Without the CSB, all sub-systems (both hardware and software) within the architecture would require a direct connection, including myriad extraction, translation, and loading routines. For n sub-systems in such an architecture, one would need

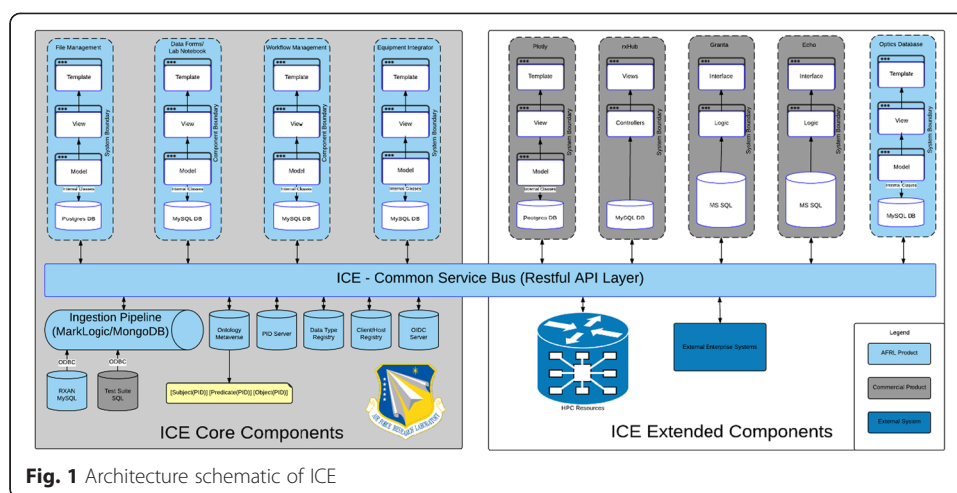


Fig. 1 Architecture schematic of ICE

$(n * (n - 1))/2$ connections in order for each sub-system to communicate with one another. Since components of the architecture are commercial packages, any changes to interfaces via the commercial vendor would result in the redevelopment of $n - 1$ interfaces.

The CSB utilizes a robust RESTful API developed in Python using the Django model-view-controller (MVC) framework, which provides controlled access to critical data models, datasets, and digital objects residing in all of the ICE sub-components [20, 21]. Using persistent identification (PID), a Data Type Registry (DTR), and a metadata repository based on triple stores (each described in detail later), the CSB can readily identify the location of a required item and how to retrieve it. Without having this level of total ecosystem awareness, sub-components would need very specific and customized integration logic. In ICE, a sub-component only needs to know how to speak to the CSB—the CSB will initiate all transactions according to the needs of all participating sub-components. This approach allows ICE to be implemented as a truly federated architecture. As ICE is intended to be a platform for internal research collaboration, federation in terms of establishing bi-directional exchange of information has been limited to internal components that remain behind an organizational firewall. However, the architectural design of ICE permits any level of federation. As an example, the ICE Search function has been successfully federated with the University of Michigan's Materials Commons through each platform's RESTful API [22]. Researchers now performing a query through ICE's Search function will automatically be searching the Materials Commons data repository as well. In this case, the federated search only benefits users within ICE, as the Materials Commons cannot contact ICE directly in a bi-directional search scenario. However, efforts are being made to allow such searches to reach end points within segments of ICE from outside the organizational firewall.

Data storage

Since the data is being generated from a variety of data sources, the sizes of and use cases for the different data sets varies widely. Some data sets are small and accessed on a regular basis, but others are 40+ TB and are accessed on an infrequent basis. Due to this diversity in data quantity and access frequency, ICE does not require aggregation into a single data storage medium due to the challenging trade-offs among storage speed, capacity, and cost. Also, to minimize conflicts in the system and maintain a

single record of reference, a data object is only stored in one location within the federated storage. This particular approach de-conflicts typical master data scenarios by (1) using system-wide unique identification and (2) establishing links to data sources rather than replicating them. By utilizing a federated data storage solution, retrieval times, storage media, and backups can be varied to the individual data requirements. Additionally, data storage solutions that already exist in the laboratory can be integrated with minimal intrusion into the underlying data structures.

Persistent identification

In order to adequately identify all objects (files, datasets, users, software, etc.) that are contained in the ICE ecosystem, a PID system was implemented. PIDs are simple universally unique identification (UUID) strings that intend to provide all the necessary information to retrieve the data they represent. Each PID is assigned to a particular client location, along with its local unique identifier (e.g., a primary key and a globally unique identifier (GUID)) and/or a Uniform Resource Identifier (URI). Any sub-component that needs to access the PID can then simply call the CSB and retrieve the data. By policy, any item created within the ICE ecosystem is assigned a PID. Other core systems, such as the ICE metadata repository, can then connect and combine data from throughout the ecosystem.

Further, participating sub-components can reliably use the ICE PID as a valid local GUID. In this way, representations of physical samples and digital items that exist in any of the participating sub-components can be joined into a hierarchy or a timeline, establishing a complete view of material provenance through the federated CSB. Any system can make a request to the CSB in order to retrieve a PID and the underlying object, regardless of its location. Thus, material composition, processing, characterization, testing, simulation, and any further data generated from within ICE are traceable to their very inception.

Metadata and data types

The ICE metadata repository (referred to locally as the “Metaverse”) is designed to maintain a complete description of the attributes of each PID, regardless of the digital item’s location. For example, the experimental data from a tension test may be stored in a data system outside of the ICE content management system. In addition to the PID and location of the data (e.g., .csv file with ID X located in system Y), the Metaverse identifies certain key properties of the file (such as the assertion that it contains “experimental data” for a “tension test,” for example). These assertions are contained in a “triple” statement which follows the form [Subject] [Predicate] [Object], yielding entries such as [Sample X] [Has Property] [Elastic Modulus]. Each part of the triple statement is assigned a PID for ease of retrieval and is indexed for performance. Additionally, ICE employs a Data Type Registry (DTR), which maintains “classes” for the various data objects in the ecosystem. A given class will identify the properties associated with each instance of the class, for example, the DTR would assert that plastic elongation is a property associated with tension test results. These properties are used to inform both the creation and retrieval of data throughout the ecosystem and form the basis for an ontological approach to knowledge management and the integration of Semantic Web technologies [23].

Search

Through PIDs, the DTR, and the Metaverse, ICE is able to provide powerful search functionality that enables a user to retrieve data regarding any object within ICE. The ICE Search engine consists of a lightweight interface and a RESTful API, which can be used by the Search front end or any other service within ICE and is enabled by the CSB. In creating a generic search end point, the ICE CSB enables any service to search for any data within ICE quickly and efficiently. The API uses metadata stored in the DTR and the Metaverse in order to identify objects that meet the criteria defined by the search terms. In some cases, secondary attributes might be identified by the search query as being relevant. For instance, the query [sample with optical spectra > 0.25 microns and refraction index = 3.10] identifies a specific subclass of a general concept of “sample,” trawling through the known data types to rapidly find those related to “sample” which contain certain attributes. Upon selecting the objects, links to the objects themselves are then discovered by querying the PIDs. These results are considered “primary results” by the API. These primary results feed into the discovery of “secondary results.” The secondary results are populated by identifying objects within ICE that have a first-degree relationship to the primary result, whether within the host system or in any other sub-component. The Search API can therefore be used to enable user-friendly traversal of material provenance, tracing an object and its relationships to other objects, as shown in Fig. 2. Latency is a particular concern during Search and can be affected by many factors including extrinsic factors such as network speed. ICE works to minimize latency through use of a RESTful API interacting with well-defined data models and well-indexed metadata of digital objects. The ICE development team has partnered with suppliers of data repositories (both COTS, open source, and internally developed) to implement agreed-upon protocols that ensure both performance and accuracy.

Collaborative group and project spaces to share data, research notes, tasks, and visualizations

As the HUBzero® platform was designed to support scientific collaboration as its primary purpose, it contains a number of features that inherently provide this functional

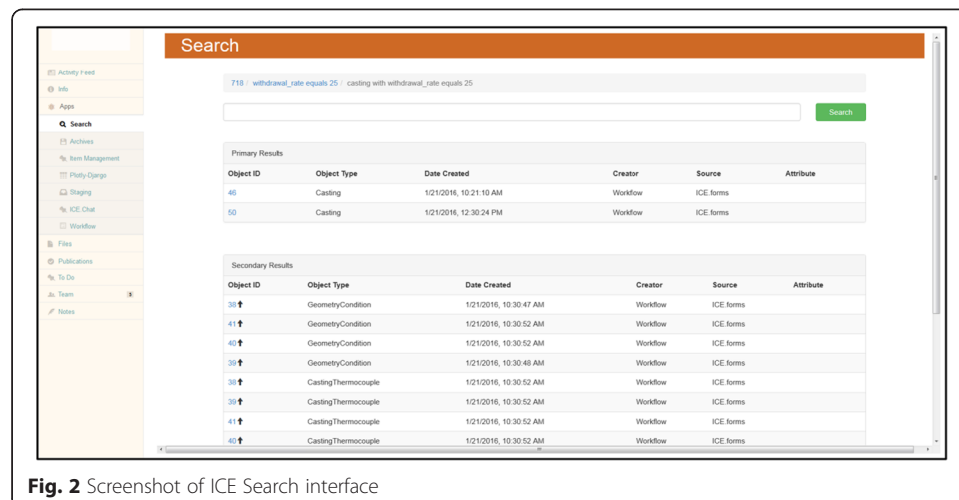


Fig. 2 Screenshot of ICE Search interface

requirement. Users can establish social profiles, research groups can be formed, and projects can be created for research collaboration [24].

HUBzero's® RESTful API provides a convenient interface through which all supporting sub-components can be seamlessly accessed via the CSB in one location, allowing the user to easily transition between applications and databases. Current applications include an equipment integrator, a sample/material management system, visualization tools, and a chat/message board application. Additionally, a robust document management system (DMS) provides access controls, document versioning, and dynamic meta-data scaling via key-value pair assignments. To date, all non-HUBzero® components have been developed using the Django framework.

Workflow management

Workflow management allows researchers to create and execute dynamic and flexible scientific workflows containing large numbers of participants in a complex flow. All ICE tools are complemented with a graphical workflow management toolset also developed in Django. This toolset employs a simple set of basic activity types—workflows (process “containers”), processes, and decisions. By allowing the user to specify sequencing, one-to-many processes, feedback loops, and nesting of an unlimited number of sub-workflows, nearly any research process scenario can be modeled. Figure 3 shows a notional example of the workflow interface. Processes that are executed in the workflow tool use a rich library of data collection forms, which are linked closely to the DMS, DTR, and metadata repository to ensure traceability of all data collected. For example, a casting process would include a form for identifying all relevant properties, from thermocouple placement to withdrawal rate. Such a form is shown in Fig. 4. The data forms (and many other components of ICE) support structured and semi/unstructured data by way of a MongoDB instance [25]. MongoDB represents the largest market share of unstructured database formats, which are critical for creating many thousands of dynamic data structures during run-time (relational databases in an MVC architecture require code releases in order to modify any part of the structure). Tying together so many key features in the workflow toolset provides the following:

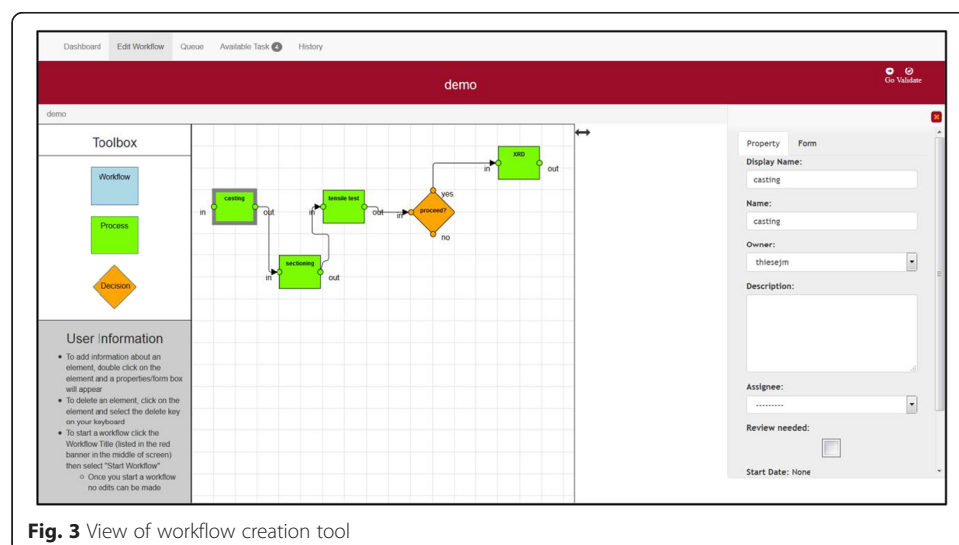


Fig. 3 View of workflow creation tool

Casting: Inputs

Owner: jacobsmid

Process Short Code: DS

Mold Configuration:

Input Items: × × + create new item

Mold Configuration File: Browse

Processing Conditions:

Mode: × +

Withdrawal Rate: Units:

Furnace Temp: Units:

Alloy: × +

Alloy Weight: Units:

Thermocouples: Type Location Number × +

Heat Treatment:

Geometry Conditions:

Name	Pattern Short Code	Attributes	Seeded	Notes to Operator

Post Cast Analysis:

X Ray:

FPI:

Laue:

Fig. 4 Example data collection form

- Complete material provenance, from raw constituents to finished materials.
- Tracking of all material inputs and outputs related to candidate research activities.
- Data and metadata from each research instance—this includes all parameters and measured data for material synthesis/processing, test, characterization, modeling, and simulation.
- Material properties and characterization results for all samples and specimens. This may include structured, semi-structured, or unstructured data.
- Comprehensive data lineage—this must include all activities relating the material data and metadata.
- Enhanced repeatability and reproducibility of the modeled process.

Seamless collection of experimental and simulation data from equipment, operators, and software

A key feature of the cyberinfrastructure is an ability to directly link experimental equipment to the laboratory intranet and then to ICE, which is implemented within ICE in several ways. First, a common file server was created that can be accessed by the research equipment. ICE employs a Staging web service that scans this “drop box” at a frequent interval; when a new file is discovered, it is assigned a PID and other metadata before being placed in the DMS for future retrieval by a practitioner at any network-connected workstation. A second method of collection is accomplished via Sweep, a Java-based client application. Sweep allows users to set up rules for watching local repositories for new data files. When a rule is triggered, Sweep behaves much like Staging as it feeds the files and metadata to the ICE DMS and metadata repository. Lastly, any equipment-controlling software toolset can be configured to interact with the DMS via

the CSB to store new files. For example, ICE interfaces with the software integration layer developed by MTS Systems Corporation for use with laboratory equipment Echo™ [26]. This capability will allow workflow managers to not only create processes that call the Echo toolset during the workflow but also define what data are to be collected by the user and Echo.

Integrated access to modeling code, data analysis, and high-performance computing resources

In providing a method for users to publish their customized modeling or analysis toolsets to the larger community, and maintain semi-autonomy, ICE does not attempt to absorb or suppress these toolsets. The Rappture and Pegasus tools resident within HUBzero® provide users with the capability to publish their custom code and tools to the rest of the community while leveraging high-performance computing job queueing and load balancing. Additional “feature” sub-components include Plotly [27], a recent but powerful addition to advanced data analysis and visualization, and Dream.3D/SIMPL [28], a toolset used for management, analysis, and visualization of hierarchical spatial data. Several other candidate components are currently under evaluation.

Single sign-on, identity management, and role-based access controls to data and components

The primary component for ICE authentication and authorization protocol is OpenID® Connect, which gives strong security while still providing single sign-on (SSO) capabilities and providing the gateway to the CSB [29]. For some ICE components where no authentication layer exists, simple Python and JavaScript clients have been developed to communicate with the OpenID® Authentication server. All members of the ICE community possess a PKI-enabled ID card, which will serve as a personal and verifiable certificate for authenticating to ICE. This authentication schema is essential for allowing ICE sub-components and users to communicate. With few exceptions, calls to the CSB or any other API in ICE will require an access token granted by the authentication server. Because this requirement applies to the creation, modification, deletion, or reading of data, ICE is able to track versions and apply access controls for all data objects down to the field/attribute level.

Challenges

In the current era of ubiquitous and networked computing, numerous challenges arise when architecting and implementing a system of the breadth and complexity of ICE. These challenges can be loosely grouped into the following categories: acquisition approaches, federated integration, master data definition, and cybersecurity. These challenges are not unique to the ICE development effort, but their specific manifestations warrant detailed examination.

As a federated system, ICE exists on the premise that a monolithic design approach cannot meet the needs of the organization. Instead, it is the coupling of tailored and specialized sub-components through a common interface (the CSB) that offers the greatest amount of useful and adoptable functionality to the research community. Realizing this significant benefit is not without trade-offs. In order to couple each component of the ICE ecosystem, communication protocols must be developed, data models

must be exposed and mapped, and common object identities (for users, data, and systems) must be implemented. For open-source sub-components, this effort is sometimes time-consuming but generally straightforward. However, for legacy and/or COTS sub-components, the models and object identities may not be readily available through modern mechanisms such as RESTful APIs. Integration with such systems requires either COTS or custom-developed “middleware” solutions. In spite of such a mitigation strategy, this scenario poses an ever-present challenge to the expansion of ICE, as new candidate sub-components are identified regularly.

Even in the most successful cases of sub-component connectivity, the pervasive issue of master data definition inevitably arises. By allowing relative self-governance among the various members of the ecosystem, record-of-reference conflicts are introduced. For example, two conflicting records may exist which describe the same physical sample. In such a case, a programmatic resolution is inadvisable; proper arbitration requires engagement by subject matter experts, which is very time-consuming. In order to avoid these types of conflicts, object classification and description is employed wherever possible. In this way, ICE becomes “self-aware” and is able to provide a high degree of master data governance. It is worth noting that in order to achieve any degree of object classification, domain experts must perform an exhaustive definition of their subject matter, which is then contained in the ICE DTR and metadata repository. However, such an effort requires extensive “treaties” to bind those objects that an expert may define for the entire ecosystem (for example, only one domain may define a turbine blade).

Cybersecurity issues have received increasing amounts of attention from the leaders of virtually all organizations. In essence, these issues concern achieving a delicate balance of security/sanitization and throughput. The most secure system design is one where every component is both logically and physically isolated. Conversely, the most usable and performant system design is one where full integration is achieved through complete component and data model exposure. In the case of ICE, the hybrid architecture calls for the harmonization of vastly different behaviors among its COTS and open-source sub-systems, while bearing the previously mentioned design approaches in mind. To meet this challenge, security policies that define such areas as data exchange and user authentication/authorization must be centrally implemented and managed. As an example, even if a particular toolset is equipped with a robust authentication system, it must be compliant with ICE protocols in order to be coupled with the CSB. To be certain, this approach adds time and cost to the development effort, but ultimately, it is the only way to navigate the myriad security requirements of a federated architecture.

While there are many other challenges that could be discussed, a central challenge in this effort has been the trade-offs between choosing an infrastructure technology that allows for highly customized development or one that is readily available in the form of a COTS software package. For reasons of speed of implementation, perceived stability, and long-term support, a COTS solution can be seen as the best path to realizing an ICE architecture. However, a one-size-fits-all approach is unlikely to meet the diverse requirements of materials scientists and engineers. The more customizability a product has, the more likely it will be able to meet the needs of both developers and researchers. However, developers implementing a system like ICE can become quickly frustrated with COTS products due to the restricted ability to make architecture and programming language choices. At the other end of the spectrum, a ground-up

approach allows developers to take ownership of the creation of the software infrastructure and create a custom system meeting all the customer's requirements. This approach is likely preferred by developers but is unlikely to overcome the perceived duplication of effort relative to the COTS product and can lead to higher organic supportability requirements. It can be very difficult to strike the right balance between the benefits of rapid implementation a COTS system offers against the flexibility and institutional longevity that a custom system could enable.

Conclusions

A major challenge facing the materials science and engineering community is the vast number of seams inhibiting collaboration and transfer of information between experimentalist, and modeler, between scientist and engineer, and between the materials, component design, manufacturing, and sustainment communities. In order to meet the ICME objective of accelerating materials innovation through a seamless flow and connection of information, it is becoming increasingly important for materials scientists and engineers to stimulate the development of supporting materials cyberinfrastructures. Correspondingly, developers and administrators of materials collaboration infrastructures must become engaged in understanding the research being performed in order to develop customized solutions in complex IT environments.

The development of cyberinfrastructures such as ICE is a key step to breaking down barriers and enabling the practice of ICME. These systems must have a primary goal of eliminating the seams that currently exist in the materials life cycle. A system architecture has been described that aims to eliminate these seams through the creation of a cyberinfrastructure that is responsive, modular, flexible, and extensible. A means of connecting researchers with experimental equipment, simulation code, high-performance computing resources, and data repositories through an integrated platform has been developed.

Acknowledgements

The authors are grateful to a number of individuals at the Materials and Manufacturing Directorate of the Air Force Research Laboratory for invaluable input on the design and functionality of ICE including Jon Miller, Andy Rosenberger, Zlatomir Apostolotov, Virginia Meeks, Robyn Bradford, Geoffrey Frank, Hilmar Koerner, Eddie Schwalbach, and Joel Murray. Support for this effort was provided by the Materials and Manufacturing Directorate of the Air Force Research Laboratory.

Availability of data and materials

Code supporting the central service bus, workflow management, persistent identification, and Data Type Registry are available on request from the authors. These tools will be posted in the near future on a Git-based source code repository.

Authors' contributions

MDJ was chief architect of ICE. JRF was the primary developer of the CSB, equipment connectivity, and sub-component integration. KMP was responsible as the primary developer of the PID server, DTR, and generator and Metaverse. EAW analyzed individual research requirements and provided design documentation to the development team. MDB provided significant design and functionality input as well as tested key functions of ICE. BJF managed connectivity with experimental equipment. CHW was overall project manager. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Materials and Manufacturing Directorate, Air Force Research Laboratory, Wright-Patterson AFB, OH 45433, USA. ²RCF Information Systems, Inc., 4200 Colonel Glenn Hwy, Suite 100, Beavercreek, OH 45431, USA. ³Southwestern Ohio Council for Higher Education, 3155 Research Blvd., Suite 204, Dayton, OH 45420-4015, USA.

Received: 29 December 2015 Accepted: 22 June 2016

Published online: 08 August 2016

References

1. Committee on Integrated Computational Materials Engineering (2008) Integrated Computational Materials Engineering: a transformational discipline for improved competitiveness and national security. National Academies Press, Washington, DC
2. The White House (2015) The Materials Genome Initiative for industrial competitiveness. <http://www.whitehouse.gov/mgi>. Accessed 17 Dec 2015
3. Gibbon GA (1996) A brief history of LIMS. *Laboratory Automation and Information Management* 32:1–5. doi:10.1016/1381-141X(95)00024-K
4. Taylor KT (2006) The status of electronic laboratory notebooks for chemistry and biology. *Current Opinion in Drug Discovery & Development* 9(3):348–353
5. Hanisch RJ, Berriman GB, Lazio TJW, Emery Bunn S, Evans J, McGlynn TA, Plante R (2015) The virtual astronomical observatory: re-engineering access to astronomical data. *Astronomy and Computing* 11B:190–209. doi:10.1016/j.ascom.2015.03.007
6. Klimeck G, McLennan M, Brophy SP, Adams GB III, Lundstrom MS (2008) nanoHUB.org: advancing education and research in nanotechnology. *Comput Sci Eng* 10:17–23. doi:10.1109/MCSE.2008.120
7. Goff SA, Vaughn M, McKay S et al (2011) The iPlant Collaborative: cyberinfrastructure for plant biology. *Frontiers in plant science* 2(34):1–16. doi:10.3389/fpls.2011.00034
8. McLennan M, Kennel R (2010) HUBzero: a platform for dissemination and collaboration in computational science and engineering. *Comput Sci Eng* 12 48(2):48–53. doi:10.1109/MCSE.2010.41
9. NEEShub (2015) <http://nees.org/>. Accessed 17 Dec 2015.
10. Kuriyan K, Catlin AC, Reklaitis GV (2009) pharmaHUB: building a virtual organization for pharmaceutical engineering and science. *Journal of Pharmaceutical Innovation* 4(2):81–89. doi:10.1007/s12247-009-9061-7
11. University of Illinois. (2015) KnowEnG. <http://knoweng.org/>. Accessed 17 Dec 2015.
12. MyGeoHUB (2015) <https://mygeohub.org/>. Accessed 17 Dec 2015.
13. Cebon D, Ashby MF (2006) Engineering materials informatics. *MRS Bulletin* 31:1004–1012. doi:10.1557/mrs2006.229
14. Ward CH, Warren JA, Hanisch RJ (2014) Making materials science and engineering data more valuable research products. *Integrating Materials and Manufacturing Innovation* 3:22. doi:10.1186/s40192-014-0022-8
15. Boyce DE, Dawson PR, Miller MP (2009) The design of a software environment for organizing, sharing, and archiving materials data. *Metall and Mat Trans A* 40(10):2301–2318. doi:10.1007/s11661-009-9889-y
16. University of Michigan. (2015) PRISMS project. <http://prisms-center.org/#/home>. Accessed 17 Dec 2015.
17. University of Illinois. (2015) T2C2: Timely and Trusted Curation and Coordination. <http://t2c2.csl.illinois.edu/>. Accessed 17 Dec 2015.
18. Kleese van Dam K, Carson JP, Corrigan AL, Einstein DR, Guillen ZC, Heath BS, Kuprat AP, Lanekoff IT, Lansing CS, Laskin J, Li D, Liu Y, Marshall MJ, Miller EA, Orr G, Pinheiro da Silva P, Ryu S, Szymanski CJ, Thomas M (2013) Velo and REXAN—integrated data management and high speed analysis for experimental facilities. In: 8th IEEE International Conference on eScience 2012. IEEE Press, Washington, DC, pp 1–9. doi:10.1109/eScience.2012.6404463
19. Carey NS, Budavári T, Daphalapurkar N, Ramesh KT (2016) Data integration for materials research. *Integrating Materials and Manufacturing Innovation* 5:7. doi:10.1186/s40192-016-0049-0
20. Fielding RT (2000) Chapter 5: representational state transfer (REST), Architectural styles and the design of network-based software architectures (Ph.D.). University of California, Irvine
21. Django Software Foundation (2016) Django project. <http://www.djangoproject.com>. Accessed: 17 Dec 2015
22. Puchala B, Tarcea G, Marquis EA, Hedstrom M, Jagadish HV, Allison JE (2016) The Materials Commons: a collaboration platform and information repository for the global materials community. *JOM* 68. doi:10.1007/s11837-016-1998-7
23. Kwok Cheung K, Drennan J, Hunter J (2009) Towards an ontology for data-driven discovery of new materials. *IEEE Intelligent Systems* 24(1):47–56. doi:10.1109/MIS.2009.13
24. HUBzero Foundation. (2015) HUBzero® <https://hubzero.org/>. Accessed 17 Dec 2015.
25. MongoDB, Inc. (2016) MongoDB, <https://www.mongodb.org/>. Accessed 26 Jan 2016.
26. MTS Systems Corporation. (2015) Echo™. <http://www.mts.com/en/products/producttype/test-components/software/Echo/>. Accessed 17 Dec 2015.
27. Plotly (2015) <https://plot.ly/>. Accessed 17 Dec 2015.
28. Groeber MA, Jackson MA (2014) DREAM. 3D: a digital representation environment for the analysis of microstructure in 3D. *Integrating Materials and Manufacturing Innovation* 3(5). doi:10.1186/2193-9772-3-5
29. OpenID® (2015) OpenID Connect. <http://openid.net/connect/>. Accessed 17 Dec 2015.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com